Full paper in Proceedings of the 16th Annual NACCQ, Palmerston North, New Zealand, July, 2003 (eds) Mann, S. and Williamson, A. www.naccq.ac.nz

The UDDI Registry – A Work in Progress

Professor Iain Morrison

The University of Melbourne
and Adjunct Professor, UNITEC
iain@staff.dis.unimelb.edu.au

ABSTRACT

As international standards setting processes strengthen and converge, technologies fundamental to distributed systems architectures, such as XML, SOAP and WebServices, are being introduced into the marketplace. These facilitate the development of e-Commerce models and technology deployment patterns that allow both the closed, traditional e-hub and 'open' trading partner identification and application integration on-the-fly (dynamic invocation).

To leverage these technologies efficiently, one needs to go beyond the 'discovery' phase of trading partner selection to efficient and possibly dynamic engagement and transaction phases at the systems integration and technology levels.

In this paper, we look at one recent initiative designed to support the above, the Universal Description, Discovery and Integration Initiative (UDDI). We review its design and current use profile and comment on its prospects for achieving its aims

1. INTRODUCTION

The UDDI(1) (Universal Description, Discovery, and Integration) Registry aims to accelerate the use of B2B (and possibly B2C) e-Commerce over the Internet.

In order to exchange information, select goods&services and transact, consumers and businesses (or businesses and other businesses) need to locate each other, identify the transaction and execute it. This can be done partially or totally online, with the (direct or indirect) assistance of online information collections/databases.

The main steps (search, locate, engage, transact) have historically been split into a 'discovery' phase and then a transaction phase. We are used to using directories such as the White and Yellow Pages either offline or via Browser, to start the search for a trading partner. We may play telephone tag, or exchange emails (White or Yellow Pages, Web or LDAP(X.500)), to reach a decision to purchase, then we go through another process of ordering, then order fulfilment and settlement. Each normally requires use of different registries/directories or the movement from online to offline operation and back again.

UDDI Registries therefore need to provide both a service discovery platform (advertise, search, locate) similar to the White and Yellow Pages and a service technology discovery and activation platform. They need to resolve queries about which companies (contact information – name, locale) have which services (products) available through which accessible technical means (phone, FAX,..., XML messaging, ..., Web Services).

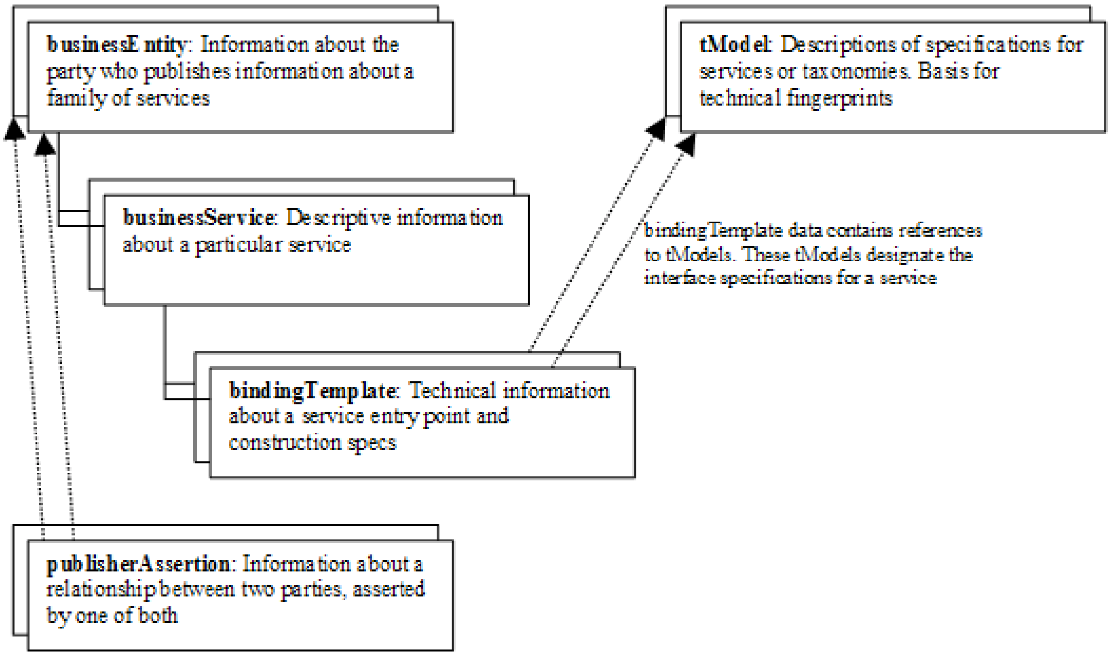

businessEntity: Information about the party who publishes information about a family of services
tModel: Descriptions of specifications for services or taxonomies. Basis for technical fingerprints
businessService: Descriptive information about a particular service
bindingTemplate data contains references to tModels. These tModels designate the interface specifications for a service
bindingTemplate: Technical information about a service entry point and construction specs
publisherAssertion: Information about a relationship between two parties, asserted by one of both

Figure 1: Logical view of Information Model

2. THE INFORMATION MODEL

UDDI takes an approach that relies upon a replicated/distributed registry of businesses and their service descriptions implemented in a universal standards compliant way.

The core component of UDDI is the UDDI Business Registration, an XML-standards-compliant file used to describe a business, its services and available means of technical engagement. The Business Registration generally consists of three components: 'White Pages' including address, contacts and known industry or service identifiers; 'Yellow Pages' including services descriptions such as industrial categorisations of the services based on standard/recognised taxonomies (including product standardised catalogues) and 'Green Pages' providing technical information about how to engage with the service interfaces that are exposed by the business over the Web (and other technologies – Phone and FAX for example). The 'Green Pages' include references to specifications for Web-Services as well as support to pointers for other service access means such as file and URL-based discovery mechanisms.

The logical view of the Information Model is provided in Figure-1[(2)]. In summary terms, each Business contains a number of Services, each of which contains Technical Descriptions (BindingTemplates) that associate with each service a service access point and means to technical implementations. The technical implementations are themselves split into concrete parts associated with each service access point and abstract part that classifies or categorises the 'type' of the means of technical engagement (the where and the how and the what). Descriptions of the technical means of engagement are bound into a new record called the Technology (or T)Model. The collection of Binding Templates are the 'Green Pages' for the business entity record.

Much of what is placed in the "White" and "Yellow" categories (the Business and Service Data Model Elements) duplicates what companies already place in other registries/directories. The Binding Templates are new and provide critical linkages between Business Services and TModels.

The Green Pages are new and provide general scope for describing which technologies are used in e-Commerce and are sufficiently general to encompass a wide range of means of technical access (Phone, FAX, Email, Web, ...) through providing a service access point (phone number, URL,...), descriptor (this is a phone number,.....) and a pointer/reference to further detailed information (service initial parameter sets and TModels). In their generality, though, the Green Pages are clearly aimed at leveraging modern XML-based messaging and service invocation based on SOAP and Web Services (W3C, 2001) and also being able to incorporate 'legacy' access arrangements such as CGI, CORBA/IIOP, Java/RMI,.. and so forth.

A feature of the Data Model[(2)] is the mandatory (limited) and extensive optional fields, and the multiple language set support. The last two, taken with the multiple/repeatable nature of the fields, leads to the entire Business Entity Record (the collection of White, Yellow and Green Page information belonging to a given company/business entity) ranging in scope and size from a minimal White Pages listing containing basic name and contact information through to descriptions of multiple services, each with multiple binding templates and all expressed in multiple language sets.

Another key feature of the UDDI Data Model is to allow the White, Yellow and Green Page entries to be further adorned/described by taxonomic classifications to aid search and retrieval. These taxonomies themselves can be user provided (e.g. in a closed membership hubbing arrangement) and as they describe the meaning of constructs are themselves TModels that are imported by reference. To assist users, a number of 'core' TModel taxonomies and type-models are made available to all – such as DUNS Identifiers, type systems for user-defined TModels, standardised goods&services catalogues, industry classification schemes etc.

Each entry type in the business record (whether White Pages, Yellow Pages or TModel) can therefore have associated with it 'type-collections', called Identifier or Category Bags in UDDI parlance, that further describe the business data and allow both focussed and descriptive queries such as "...find me a business in the following neighbourhood that supplies this type of service via a Web ordering system that complies with....". These "Bags" are collections of name-value pairs associated with particular Identifier or Category spaces (taxonomies and TModels).

It is a key part of the Information Model that the TModels are not bound into the Business Entity record associated with each publishing entity, but remain separate and are accessed by reference. Two reasons for this, by design, appear compelling

(i) in order to search effectively through the Registry there must be an agreement on terms and meaning in public or to a specified user community and

(ii) in order to leverage investments in technology and get a critical mass of interacting (potentially) traders there must be commonality in the technical means of access (at least at the abstract or logical level), that is the TModel level.

The more a specific TModel is shared, the wider the potential marketplace with each subscriber/registrant only needing to identify their service access point for the selected means of access. An example would be TModel="phone", with the Registry acting as Directory Services and each subscriber having GreenPage entries with their phone number and possible local access codes (as initial parameters) unless a global E.164 address scheme were used. Web-based analyses for different use-contexts or use-types can be drawn (HTTP-GET and POST and CGI etc) (the Green Pages has a 'description' field to allow further elaboration).

As a summary view, the Information Model could be classified as technically appropriate (to the task of facilitating e-Commerce), of high capability, logically structured, extensible, and complying with good registry practice in terms of structural and type(schema) checks, and the use of taxonomies and controlled vocabularies (both predefined and user contributed). In spite, however, of being logically structured, business registration records can be very complex and issues of poor user understanding and poor data quality and information selection can arise (see later).

Since its inception (UDDI v1.0), the underlying Information Model has undergone some changes.

Version 2 (the current production version and the one used as a reference point in this paper) added a merged treatment of Category Bags in the Search API (i.e. allowing searches to span classifications in the White, Yellow Pages and TModels), then introduced a new information record called publisherAssertion (as above in Figure-1). This record was aimed at supporting federated organisations and organisations that either had multiple listings (but common data) or that had already established, for example, shared information (such as in a private trading hub). The publisherAssertion allows business relationships to be added to the information set.

These changes can be seen as responses to difficulties with consistent usage of taxonomies and classifications and also failure of the original Information Model to support business relationships (internal or external). Version 2 also allowed subsidiary private registries with limited capability that were 'owned-by' a public registry. Changes to the Information Model allowed private taxonomies to be used and this better supported closed communities and internationalisation as the original set were very US-centric.

Version-2 also introduced the element of 'checked' or 'unchecked' records in recognition of the fact that data quality in the Registry (although tightly constrained by schemas regarding structure and typing) was low. "Checked" registry entries have had their data validated against the underlying semantic models by a third-party. "Unchecked" means the data was entered by the registry user themselves. It is perhaps a reflection of the complexity of the Information Model and the lack of understanding/awareness in the user community that this protection/caveat was deemed necessary.

3. THE SEARCH AND PUBLISH API

The programmatic interface to a UDDI Registry is exposed using the SOAP protocol using XML-based message exchanges in a simple request/response model.

Version 2[3] defines 25 search messages and 15 publish messages. This messaging API can be exercised through a Browser (see http://uddi.microsoft.com/), or a Java Class Library (UDDI4J), or via basic XML-message construction over HTTP-POST (although there are limited functionality HTTP-GET calls).

The 'search' messages are aimed at allowing a wide range of discovery methods. The 'find-XXX' message set is biased towards general name or identifier or category search against Business Entity (or Business Service or TModel) records and returns a result set of those Business Entities that match the search criteria. The response data contains globally unique database keys that uniquely identify a business record of some type (the keys have meaning however only for the registry where the original publication took place 'the owning-registry'). The 'get-XXX' calls use these keys in drill down mode to extract specific and detailed record sets.

The Registry therefore allows both browse mode (browser or via API) and iterative drill-down programmatic mode. It was the intent of the designers that these 'keys' be cached or held in the using application to minimise searches and multiple trips to the registry databases.

The logical Information Model in Figure-1 is deceptively simple and belies the detail necessary in the search process which in all but the simplest query can involve multiple interrogations/message exchanges. The output sets generally consist of a collection of hierarchical information nodes structured as in a Document Object Model (DOM)[4] tree. A search then maps into (in the above representation) a tree-walk where the nodeset may be dynamically constructed through registry calls. Each business registered contains a complete Business Entity record within which are contained (amongst other information) a collection of Business Service records. These in turn contain collections of BindingTemplate records which contain TModelInstance elements which reference applicable TModels. In addition to these nestings of collections, each record type contains its own data about the Business, Service and Bindings respectively. This data may be KeyedReferences to TModels that either identify or classify/adorn the data with name and type semantics in some Namespace or taxonomony.

If one looks at the API details for software that implements UDDI Version 2 (such as UDDI4J or Microsoft UDDI SDK), one can see the information nestings exposed as class hierarchies or as DOM structures – providing a choice as to how the information is to be 'harvested' and the information tree walked. The APIs map the SOAP-XML messages from and to XML to DOM structures to Class hierarchies.

For the casual user, the Web Browser interface (Microsoft, IBM and HP at least provide this) is by far the easiest to use. The user specifies a partial name or wildcard (Business or TModel), selects any search qualifiers (i.e. by location, industry or service type, means of access type) and gets a result set (hopefully). The user can then go through this result set drilling down in detail until a candidate set is contained that matches requirements.

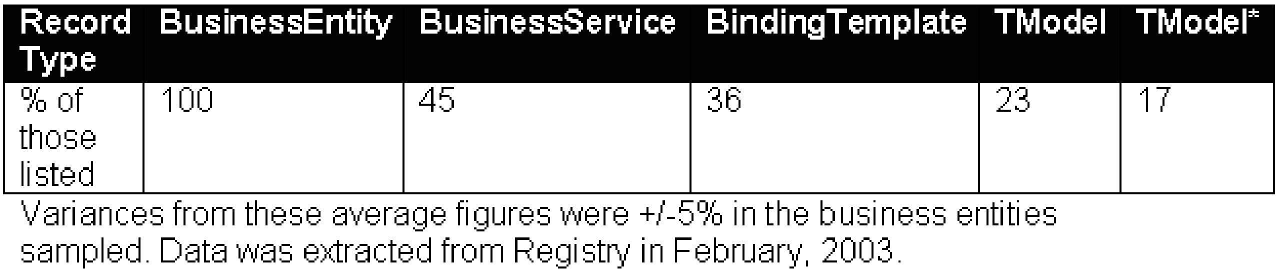
The basic SOAP messaging and programmatic APIs were constructed, however, to facilitate direct system-to-system engagement for e-Commerce (putting together search/locate/engage/transact) as effectively a single operation. This provides functionality similar[5] to CORBA's Trading Service and Dynamic Invocation which have been in place for several years (albeit not carried by HTTP nor XML-based).

4. USE AND UTILITY

It is instructive at this stage to move from analysis of the Information Model and Architecture to a study of how the UDDI Public Registries, which have been 'open' for ~ 2 years, are currently being used.

One question to ask is how many large public companies are making use of, or experimenting with, the UDDI Public Registry. Taking the Fortune 500 membership (in Australia the ASX200 membership) and querying the Registry leads to the result that only ~1% of US public listed companies have a presence and 0% of ASX membership. It should be noted that in conducting this data collection, any evidence of business entity presence was accepted (White or Yellow or Green Page data). Our results included companies even if it were evident that the Green Pages elements were missing or had service access points that were non-operative (that is were non routable IP

Table 1: Records by Type



Record Type	BusinessEntity	BusinessService	BindingTemplate	TModel	TModel*
% of those listed	100	45	36	23	17

Variances from these average figures were +/-5% in the business entities sampled. Data was extracted from Registry in February, 2003.

addresses or 'localhost') as appears common in exploratory mode.

Reversing the direction of analysis, one can ask what types of business entity are represented in the Registries and what types of use is being made of the capabilities of the technology.

We conducted extensive sampling of the Registry data by the name of the business entity, drawing 10 initial letters ('a', 'b', ...) at random (case insensitive) and conducting analysis of ~50% of business entities listed as having names beginning with that initial letter.

Our analysis measured the number of companies listing Business Services (all companies of course had a Business Entity Record), the number who then listed Binding Templates (associating listed services with specific technological means of engagement) and then who used specific types of TModel. In the latter case, we used the core types taxonomy as a classification for identifying those offered through Web, Soap, XML, WebServices/WSDL, Transport or Specification.

In Table 1, we summarise the range of results found.

The results show that the majority of businesses registered provide effectively a White Pages entry only, giving company details, contacts and (possibly) some information via 'Bags' on the company services or industry type via the Universal Services and Products Catalog (UNSPC) or the North American Industry Classification Scheme (NAICS).

Just less than half (~45%) provide an entry that contains descriptions of available Business Services/ Goods (most commonly by far at 1 per business entity), and of these, about 80% (i.e. 36% of the total) proceed to list available technology channels by providing TModelInstance data. Again about 80% of this Instance data makes reference to technology channels at or above Web-based capability. Very few, at present, claim to be WebServices capable.

These latter figures are in themselves deceptive in that in approximately half the cases the service was incapable of activation because of either

(i) no or inaccessible Access Point

(ii) incorrect TModel linkage in the Binding Templates

(iii) improper use of TModels

So the proportion of entries in the Registry fully capable of exercising its purpose/capability through discovery through transaction, we assessed at less than 10%. All registrants were using the Registry for White Pages only with just under half of these providing further service information. It was clear that there was some exploration of the Green Page and TModel capabilities by registrants but this was being done on a 'live' Production Registry rather than available Test Registries.

In terms of the types of company using the Registry, our initial approach was to try to analyse the Category Bags, that is the identifiers and descriptions added by registrants to more fully describe their company and goods and services. We found this difficult for a few reasons:

(i) few companies bothered

(ii) inconsistent use of NAICS and UNSPC classifiers

Relying more generally on the 'Description" fields, it appeared that those registering were predominantly small private companies in the information, financial, technology or software services sectors. This result is consistent with the (self-selecting) client survey conducted by technology-provider Systinet[(6)]

In terms of utility, we can only comment on the difficulties experienced in conducting our search and analysis on a Version 2 Registry.

Firstly, although the use of taxonomies and Category Bags prima facie helps search precision, the lack of a meaningful convention for their use in the context of the search (for example you are expected to specify the 'exact' identifier obviating the clear drill-down mechanism in the industry sector or products classification tree) leads to poor selectivity and query resolution. The choice of specific taxonomies is also not helpful as these are either jurisdiction specific or unknown to the requestor. Keywords may be difficult to use in a multinational/multilingual context such as the UDDI Registry however some assistance with search space definition or ontological mapping is indicated (Reference). Some publishing Business Entities tried to use multiple taxonomies and multiple classifiers/language sets to make their entry more visible.

The same type of criticism can be addressed in Version 2 at the Green Pages/TModel level. Searches can specify specific TModels to search for (by database key!!), but not 'types' of TModel. This makes it indeed very clumsy to find business partners with accessible integration pathways (you effectively have to know the answer before you ask the question). This shortcoming is being addressed (see below) in a revision of the specification. Although there are elements in the Data Model that could assist, these are not mandatory, are free-text not controlled vocabulary, and appear poorly understood and poorly used (if at all).

5. DATA QUALITY, CONVENTIONS AND SEMANTICS

Although the Information Model and associated Data Elements and schema tightly control the structure of information, the current utility and utilisation of the Registry is, in our view, being hampered by a number of factors.

The open-endedness of publishing access and non-authentication and validation of data allows many spurious entries to be generated, lowering search efficiency and selectivity and no doubt furstrating many users.

The balance and nature of the controlled vocabularies (as instanced by the use of taxonomies) as against free text descriptions is such that it is difficult to ask 'natural' questions. This may of course be more of a problem in browse/search mode rather than a later engage mode where the enquiror may have already constructed a 'pool' of available and compatible business partners.

Data quality would be improved through more use of controlled vocabularies in key contexts (such as the Green Pages) and in a clearer articulation of conventions and semantics. In the case of WebServices and WSDL, for example, there were initial difficulties in cleanly separating the abstract/logical technology description (aimed at the TModel) from the concrete implementation detail (aimed at the Green Pages). The semantics did not fit easily into the BindingTemplate/TModel hierarchy and spilled over into further reference material provided through the actual TModel itself. This was partly addressed through a 'convention' introduced[7] , but does not appear to have been very effective. Again the Version 3 specification will attempt to address this problem through introduction of new data elements as WebServices/ WSDL was a key driver for the project but this will leave a large body of Version 2 data which is impaired.

In terms of the data itself, its use and meaning, the information model is deceptively simple but allows a multitude of ways to say the same thing. There is little uniformity in approach and a lack, in the public domain, of articulation of conventions. There also appear difficulties with understanding the meaning of key data elements in context (the multitude of Category Bags, for example was confusing and this led to the 'merging' of Bags for search purposes in Version 2 to try obviate the effect of the confusion on the search process result space). Better advice on how to use, or tighter control on, the identifier and type spaces would also help.

There are a number of approaches which can help:-

(i) the introduction of the 'checked' and 'unchecked' entry qualifiers to signal external quality control on a business entity's records

(ii) the use of private, associate or affiliate registries, linked to specific use communities and the main Registries, but with tailored and constrained use profiles

(iii) introduction of intermediaries to profile a company, on a commercial basis, onto the Registry and restrict access

(iv) strengthening controls within the Registry Information Model and API itself to reduce the occurrence of semantically incorrect and inconsistent data.

(v) industry-wide profiling

Currently initiatives are underway in all of these areas to try to address the problems and increase confidence in the technology and its application. 'Checking' is already available as is access to commercial profilers. The Version 3 Specification addresses some of the others. The industry-wide profiling can be through initiatives such as WSIL[8] which seek to define simplified TModel sequences for adoption, or sector specific initiatives which seek to build communities of use in specific application areas with their own controlled vocabularies, policies and TModel sets.

6. CHANGES

In response to some of the difficulties and shortcomings raised above, the Version 3 Specification for UDDI has just been released. There are as yet no implementations.

Version 3 addresses some further perceived shortcomings in the Information Model, The Registry Model itself and usage patterns to date.

The original Registry Model assumed that each business selected an 'owning' UDDI Registry which then replicated the business record to other 'public' registries (there have been usually 3-4 of these). There were issues with the generation of identifiers for elements of the business record as these were locked to the 'owning' registry thereby obviating transfer of record between registries and allowing stale keys on using sites/applications. UDDI-based applications therefore lost location transparency. The key formats themselves in the Information Model are proposed changed to a form that uses more acceptable URL-style keys and obtains better transparency.

The generality and complexity (and data quality) problems with the 'core' public registries also caused problems for using groups that that were wishing to establish communities through the Registry. In Version 3, the Registry Model has been extended to be much more like the DNS Information Model and Implementation in allowing more devolution to hierarchical subordinate registries, less centralised control, information migration, policy mechanisms and inter-registry communication policies and procedures. A critical omission in Version 2, namely the failure to allow BindingTemplates to be adorned with CategoryBags (i.e. having descriptions of the technical means of access supported being directly accessible in the Business Record) is proposed corrected in Version-3. Support for Digital Signatures and record administration (time stamps etc) is to be added to allow Business Records to be authenticated – a major security hole in the Registry to date being that there was no check on the publishing entity allowing masquerading and hijacking of e-Commerce identities.

7. BUSINESS MODELS AND COMMUNITY

The business model for the operation of the Registries themselves is still open in that the main proponents (IBM, Microsoft,..) have reserved their position in terms of access and licencing of technology. There are however a small number of public domain implementations in whole or part.

At this stage of the technology emerging, it is in the interests of the major stakeholders to encourage experimentation and use, on a free basis, and for all parties to benefit from the interactions and learning experiences.

As noted above, the original model of a large monolithic public registry that attempts to be all things to all prospective users is changing as the realisation build that this, in its original use context, leads to increasing complexity and poor data quality and service levels.

The model of a single (albeit replicated) Public Registry is moving (at least as instanced by changes mooted in Version 3) to that of a series of linked distributed Registries conforming to a single architecture. Each registry may have its own controlled vocabulary and classifications schemes, policies, restrictions on TModels and application areas. The UDDI cloud is then a mix of public and private registries interoperating in some way yet to be fully articulated but conjectured as modelled on the DNS with 'root' registries able to resolve queries to tier-1 registries then to tier-2 and so forth, with multi-level caching of result sets.

Alternate models exist. IBM has proposed[9] a fully distributed peer-to-peer system, where each corporate Website has, at a known URL, a description of the Web Services that it supports. This description is encoded in XML, is schema-controlled and is able to be read and services automatically activated in the WSIL, WSIF framework. It other words, it acts as a stripped (dramatically) Business Entity record that still facilitates a measure of discovery, engagement and transaction albeit in a limited set of TModels/technology spaces. In the IBM alternate, one would search for services in 'known' partners (perhaps those with established trading relationships) and use/cache these.

Even more simplified is the Small Business Initiative (SMB)[10]. This proposes a very small Information

Record per business entity (~20 elements mostly of free text) which can hold simple business contact, industry/service descriptions and technology access point information (which may just be a Website). These records can be centrally held and managed (like UDDI) or distributed to corporate sites (like the IBM alternate). Yahoo-style metadata driven search and retrieval (search engine) or Xpath-type document retrieval in a central database can be very efficient. Each of these variants, with their substantial simplifications, could appeal to different user communities which do not need nor want the full power and complexity of the UDDI Registry. It does beg the question though of whether these alternates will co-exist synergistically or be captured by different communities of use.

8. BARRIERS TO ADOPTION

Even if the difficulties with the Information Model and current patterns of use are overcome, there remain some impediments to widespread adoption.

One of these is the validation and authentication of company information placed in the UDDI Business Entity Record and the ensuring of properly authorised changes. At the moment there is no check on linkage between the person securing the record and change permissions and the entity being claimed represented. In addition the information provided is not validated. This exposes a wide range of security problems.

The security model for the registry itself is an issue as the Version 2 specifications left it to each Registry provider to establish their own framework and mechanism for messages with implied write access requirements.

With registry migration becoming an issue, there are dual problems in ownership of the security model and data and ease of migration. In the longer term these will be resolved through Digital Signatures, Digital Certificates and Certificate Authorities but there are many unresolved issues in this general problem space. In this regard closed or semi-closed communities of use may be able to move more quickly. Certainly endorsement and acceptance of public security frameworks is still a hot area and far from settled although there have been recent advances[11, 12].

Overlaying the above are the security models associated with the use of specific TModels (that is application-based). Given the aim of the UDDI Registry to facilitate external integration, there are a number of business process and application security issues that need faced. The TModel may invoke 'https' or 'ftps' and key-based transport-level security such as SSL. It may specify a Web Service through WSDL but still leave unresolved how the application level methods and data are to be protected. Companies would wish to avoid different security regimes and frameworks for Registry access and use and application invocation.

The recent move by IBM and Microsoft to take forward a SOAP-based security standard to W3C and OASIS, and to implement Digital Signatures in Version-3 in the SOAP-based Registry should be seen as an attempt to crystallise out decisions on some of the above considerations. Web Services Toolkits available from both companies are also offering some initial exposure to early adopters/experimenters. Certainly data integrity and authenticity could be improved by this means but the data quality and semantic/use problems still will remain in all but tightly controlled use communities.

In the absence of a rapid resolution of the public security model, businesses would be looking at the real infrastructure and systems costs of a UDDI-mediated integration project, and at the benefits that might accrue. It would appear that there is a greater chance of the technology being adopted in the circumstance of an already established and technologically mature use community that is close to being closed and where any investment in (possibly interim) security architectures and support processes may be able to be amortised quickly. Intranet (internal inter-application use) and Extranet (closed trading community) use scenarios may lead this type of uptake.

9. SUMMARY

We have reviewed the architecture and technology of the current UDDI Registry Model and assessed current use patterns against the aim of enhanced service discovery and facilitated systems integration.

The Information Architecture is at once both simple and powerful. At present, its flexibility, however, lack of key contraints and articulated conventions/profiles, and non-validated and authenticated use are hindering its uptake and perceived utility (relative to its complexity). In attempting to anticipate and accommodate all prospective business communities and potential technology solutions, has the Registry rendered itself too difficult to use effectively?

The Information Architecture and Contraint Set are being enhanced in the light of experience gained to date but alternate limited functionality, but simpler, models are emerging. These simpler alternates and current difficulties with security models are suggesting that more distributed, use-context and user-driven technology adoption might be preferred. The concepts and ideas of the UDDI Model might be appropriated into simpler use frameworks. One only has to bear in mind the OSI Profiles process and the X.500/LDAP transition to realise that, on the Internet, clever, simple solutions tend to get adopted in preference to powerful, but complex, solutions and critical mass emerge ‘bottom-up’.

The top-down technology-driven approach of the original monolithic UDDI Registry Model does not sit comfortably in this scenario. The incremental and bottom up approaches of the IBM and SMB variants may initially have some advantage in affording some exposure to key concepts and technologies but are limited in scope and scale and possibly to specific communities.

The changes in UDDI Version 3 towards a more devolved model appears to accept that the monolithic central solution had technical, use and political problems and that a bottom-up approach of a number of cooperating Registries may have greater traction. It does throw into focus, though, the overarching Information Architecture and Interoperability and Security frameworks.

Version 3 also introduces query paths tailorable on a per registry basis and nested queries which are aimed, one expects, at trying to reduce the number of search iterations and the currently poor support for TModel interrogation (the ‘useType’ field is only a part solution/hint to types of queries and some services need to invoke multiple TModels simultaneously). Is there a danger of increasing the power and complexity at the Information Model and Search API at the expense of simplicity and accessibility??

Industry pundits set the expected timeframe for meaningful Web Service/WSDL uptake as from 2004. Whether this is manifest through an, albeit distributed and devolved, UDDI Registry Model or through use communities using simpler discovery and engagement methods is still an open question. The direction of evolution of the UDDI Initiative, as evidenced in the Version 3 Specification, is that the main proponents now believe the aims might better be secured by aggregates of simpler Registries holding to the same API and Information Architecture (albeit possibly policy constrained) inside a common interoperability framework.

BIBLIOGRAPHY

1. UDDI, 2002: UDDI - Universal Description, Discovery and Integration of Businesses for the Web. Executive and Technical White Papers.

http://www.uddi.org/

2. UDDI, 2002: UDDI Version 2.03 Data Structure Reference. http://uddi.org/pubs/DataStructure-V2.03-Published-20020719.pdf

3. UDDI, 2002: UDDI Version 2.04 API Specification.

http://uddi.org/pubs/ProgrammersAPI-V2.04-Published-20020719.pdf

4. The Document Object Model (DOM). W3C Architecture Domain, July 2002.

http://www.w3c.org/DOM/

5. The Object Management Group. CORBA – The Common Object Request Broker Architecture

http://www.omg.org/, http://www.corba.org/

6. SIIA/Systinet Web Services Survey Report. 2002

http://www.siia.net/divisions/software/members/pubs/wssurvey.pdf

7. Using WSDL in a UDDI Registry. 2002.

http://www.uddi.org/pubs/wsdlbestpractices.pdf

8. WS-I. The Web Services Interoperability Organization. 2002.

http://www.ws-i.org/

9. WSIL – The Web Services Inspection Language. 2001

http://www-106.ibm.com/developerworks/webservices/library/ws-wsilspec.html

10. The SMBmeta Initiative, 2003.

http://www.trellixtech.com/aboutsmbinitiative.html

11. Security in a Web Services World: A Proposed Architecture and Roadmap

ftp://www6.software.ibm.com/software/developer/library/ws-secmap.pdf

12. Web Service Security

http://www.oasis-open.org/committees/wss/ , http://www.w3c.org/